GEQ: Gaussian Kernel Inspired Equilibrium Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite the connection established by optimization-induced deep equilibrium models (OptEqs) between their output and the underlying hidden optimization problems, the performance of it along with its related works is still not good enough especially when compared to deep networks. One key factor responsible for this performance limitation is the use of linear kernels to extract features in these models. To address this issue, we propose a novel approach by replacing linear kernels with a new function that can readily capture nonlinear feature dependencies in the input data. Drawing inspiration from classical machine learning algorithms, we introduce Gaussian kernels as the alternative function and then propose our new equilibrium model, which we refer to as GEQ. By leveraging Gaussian kernels, GEQ can effectively extract the nonlinear information embedded within the input features, surpassing the performance of the original OptEqs. Moreover, GEQ can be perceived as a weight-tied neural network with infinite width and depth. GEQ also enjoys a tighter generalization bound and improved overall performance. Additionally, our GEQ exhibits enhanced stability when confronted with various samples. We further substantiate the effectiveness and stability of GEQ through a series of comprehensive experiments.

1 Introduction

Deep Neural Networks (DNNs) show impressive performance in many real-world tasks on various data like graphs [40], images [38, 20], sequences [8], and others. However, most neural networks structure are constructed by experience or searching on the surrogate datasets [28, 54]. Therefore, these architectures cannot be interpretable and such a phenomenon hinders further development. Apart from the current neural network models, traditional machine learning methods like dictionary learning [43, 30], subspace clustering [51] and other methods [26, 53, 52, 27] can design their whole procedure by designing optimization problems with specific regularizers customized from their mathematical modeling and requirements. Thus, these models are easily interpreted. However, the traditional machine learning algorithms' whole procedures do not consider the hidden properties of features and labels. Therefore, they usually perform worse on tasks with more data.

To link two types of models, OptEqs [50] tries to recover the model's hidden optimization problem to make their model "mathematically explainable". They claim that the output features $\tilde{\mathbf{z}}^*$ (we also called the equilibrium state) with respect to input $\mathbf{x}$, which are obtained by solving the fixed-point equation in Eqn (1), is the optimal solution for its hidden optimization problem defined in Eqn (2).

$$\tilde{\mathbf{z}}^* = \tilde{\mathbf{W}}^\top \sigma(\tilde{\mathbf{W}}\tilde{\mathbf{z}}^* + \mathbf{U}\mathbf{x} + \mathbf{b}),\ \hat{\mathbf{y}} = \tilde{\mathbf{W}}_c \tilde{\mathbf{z}}^* + \mathbf{b}_c, \tag{1}$$

$$\min_{\tilde{\mathbf{z}}} G(\tilde{\mathbf{z}}; \mathbf{x}) = \min_{\tilde{\mathbf{z}}} \left[\mathbf{1}^\top f(\tilde{\mathbf{W}}^{-1\top}\tilde{\mathbf{z}}) - \left\langle \mathbf{U}\mathbf{x} + \mathbf{b}, \tilde{\mathbf{W}}^{-1\top}\tilde{\mathbf{z}} \right\rangle + \frac{1}{2}\|\tilde{\mathbf{W}}^{-1\top}\tilde{\mathbf{z}}\|_2^2 - \frac{1}{2}\|\tilde{\mathbf{z}}\|_2^2 \right], \tag{2}$$

where σ is the ReLU activation function, $\mathbf{U}, \tilde{\mathbf{W}}, \mathbf{b}, \tilde{\mathbf{W}}_c, \mathbf{b}_c$ are learnable parameters. They are trained by optimizing loss functions, such as cross-entropy loss, which are calculated based on the final

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

prediction $\hat{\mathbf{y}}$ derived from $\tilde{\mathbf{z}}^*$ as shown in Eqn (1). $\tilde{\mathbf{W}}^{-1}$ represents an invertible or pseudo-invertible matrix for $\tilde{\mathbf{W}}$, and the function f is a positive indicator function that induces to commonly used ReLU activation. The equivalence between $\mathbf{z}^*$ and the optimal solution for the hidden optimization problem (2) enables researchers to not only gain insights into OptEqs' behavior by understanding the underlying hidden optimization problem but also innovate by designing new models based on different problem formulations tailored to specific tasks. For instance, OptEqs introduces a module that promotes output sparsity, and Multi-branch OptEqs (MOptEqs) incorporates fusion modules to enhance the diversity among its branches. Despite these advancements and the incorporation of various modules inspired by different vision tasks, the performance of OptEqs and related models still falls short when compared to deep neural networks in image classification. This discrepancy suggests the existence of crucial components that limits the performance of equilibrium models.

To identify such a component, we delve into the hidden optimization problem of OptEqs and observe that it can be decomposed into two distinct parts: the regularizer term for the output features and the feature extraction term. While the feature extraction term is crucial as it depends on the input and determines the patterns extracted from the input features, the exploration of the regularizer term has been largely overlooked, with linear kernel functions being the predominant choice for feature extraction. Thereby, we believe that the limitations of previous equilibrium models stem from their feature extraction parts, as linear kernels struggle to capture complex features effectively. Building upon these insights, we take a step forward by leveraging the widely adopted Gaussian kernel for feature extraction in the hidden optimization problem.

Then by calculating the stationary condition for above new hidden optimization problem, we propose our new type of OptEqs, the Gaussian kernels inspired equilibrium models (GEQ). The model involves a new attentive module induced by its hidden optimization problem and enjoys much better performances on classification tasks even compared with deep models. Furthermore, we also prove that the new model's outputs are equivalent to the outputs for OptEqs with weight-tied "infinite wide" mappings. Therefore, an interesting finding is that our model can be regarded as a "double-infinite" model because the original OptEqs can be regarded as a weight-tied "infinite deep" model. Apart from the above findings, the utilization of Gaussian kernels also makes our proposed model enjoys tighter generalization bounds. Besides the generalization abilities, we also analyze the stability of our GEQ and find its stability is better on various inputs. We summarize our contributions as follows:

- We first reformulate the OptEqs' hidden optimization problem with Gaussian kernels and propose a new equilibrium model called GEQ. It contains a new attention module induced by its hidden optimization problem and performs better on real-world datasets.
- We find that our GEQ can be regarded as a weight-tied neural network with both infinite width and depth, and the generalization bound for our GEQ is also tighter through our analysis. Empirical results also confirm the superiority of our GEQ.
- We theoretically demonstrate the advantages on the stability of our GEQ compared with former OptEqs on various inputs. We also conduct experiments to validate such advantages.

2 Related Works

2.1 Implicit Models

Most modern deep learning approaches provide explicit computation graphs for forward propagations and we call these models "*explicit* models". Contrary to these models, recent researchers proposed some neural architecture with dynamic computation graphs and we call them "*implicit* models". A notable example of an implicit model is Neural ODEs [7], its architecture is encoded as a differential system and the implicit ODE solvers they used are equivalent to continuous ResNets that take infinitesimal steps. By representing the entire structure using differential systems, implicit models tap into the black box of traditional neural networks while offering increased flexibility and interpretability. Because of the flexibility and the interpretability of implicit models, the design of implicit models [15, 17] draws much attention these days. Many kinds of implicit models have been proposed, including optimization layers [11, 1], differentiable physics engines [37, 9], logical structure learning [47], differential programming [49, 40, 6, 4, 3].

Among the various implicit models, OptEqs [50] and its multi-branch version MOptEqs [25] stand out as they not only exhibit superior performance compared to other implicit models but also explicitly establish the relationship between their structure and a well-defined optimization problem. Therefore, exploring better equilibrium models is a promising direction to achieve more interpretable neural architectures. However, it is worth noting that while OptEqs and its variants have shown promising results, their performance is still not entirely satisfactory, particularly when compared to the deep explicit models. Recent work [46] who adopts a DEQ layer as a middle module in deep declarative networks [17] and also utilizes kernel functions in its construction. The main differences between their model is that we use gaussian kernel function to construct the optimzation problem for the equilibrium state $\mathbf{z}^*$ and our models can be used in any models as the feature extraction layer. Besides these models, other works [48, 39] also show the connection between their architectures and optimizaiton problems, but their performance are also not satisfying.

2.2 Infinite Wide Models and Kernel Methods in Deep Learning

By employing kernel methods to estimate the outputs of single-layer networks for various samples, researchers discover that such networks can exhibit characteristics of a Gaussian process (GP) when their parameters are randomly initialized with a large width limit [32]. Building upon this idea, recent researchers have extended these findings to neural networks with multiple layers [24, 10] and other architectures [34, 13]. These studies primarily focus on weakly-trained models, where the network parameters are randomly initialized and kept fixed throughout the training process except for the last classification layer [2]. Despite their "weakly-trained" nature, these models still provide valuable insights applicable to current neural networks. For instance, mean-field theory [5, 16, 19] explains phenomena such as gradient vanishing and exploding during back-propagation, which are relevant not only to single-layer networks but also to other structures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Other researchers explore stationary kernels to enhance the interpretability of neural networks by designing different activation functions [31].

In addition to weakly trained models, recent studies [22, 2] introduce the concept of Neural Tangent Kernel (NTK) and its variants. These works have demonstrated that the sample kernel of infinitely wide networks, with appropriate initialization, can converge to a fixed neural tangent kernel when trained using gradient descent with infinitesimal steps (gradient flow). The NTK model is a theoretical construct with strict constraints, and its weights are not learned. It is important to note that although our model can also be seen as an infinitely wide model, there are several key differences between our approach and the aforementioned models. Firstly, our model utilizes kernel methods to operate on input features and output features, while the NTK models employ the kernel method on samples. Secondly, our GEQ model can be viewed as employing a "weight-tied infinite wide" projection that is parameterized by learnable parameters, allowing for updates during the training process. This contrasts with NTKs and NTK-DEQ [12](an equilibrium model constructed with vanilla NTK layers), where the weights are fixed and not learned. Therefore, despite the potential overlap in terminologies used in our paper and NTK-related works, our GEQ model differs significantly.

3 Gaussian kernel nspired Equilibrium models

3.1 Formulation and Structure of GEQ

Before starting our analysis, we need to reformulate the original formulations of OptEqs' equilibrium equation (1) and hidden optimization problem (2) for convenience. We replace $\tilde{\mathbf{W}}_c\tilde{\mathbf{W}}^\top$ with $\mathbf{W}_c$, $\mathbf{W} := \tilde{\mathbf{W}}^\top$, and replace $\mathbf{z}$ with $\tilde{\mathbf{W}}^{-1\top}\tilde{\mathbf{z}}$. Then the original OptEqs' optimization problem can be reformulated as:

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \min_{\mathbf{z}} \left[\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|_2^2 - \langle \mathbf{U}\mathbf{x}+\mathbf{b}, \mathbf{z}\rangle - \frac{1}{2}\|\mathbf{W}\mathbf{z}\|_2^2\right]. \qquad (3)$$

With the new formulation, we can rewrite the equilibrium equation for OptEqs with input $\mathbf{x}$ by calculating Eqn (3)'s first order stationary condition $\nabla G = \mathbf{0}$ and then reformulate it as follows:

$$\mathbf{z}^* = \sigma\left(\mathbf{W}^\top\mathbf{W}\mathbf{z}^* + \mathbf{U}\mathbf{x} + \mathbf{b}\right), \qquad (4)$$

where σ is the ReLU activation function, $\mathbf{U}, \mathbf{W}, \mathbf{b}$ are learnable parameters trained by optimizing loss functions (like cross entropy loss). From problem Eqn (3), one can see that the GEQ's outputs try to extract features by minimizing the similarity term with the input feature $\mathbf{U}\mathbf{x} + \mathbf{b}$ through a linear kernel function with some constraints defined in its regulation terms to prevent the trivial outputs. Such an explanation can also extend to other DEQs [3, 48] under the symemtric weight constraints. However, linear kernel functions cannot perform well when processing complex inputs as other traditional machine learning mechanisms show. We deem that this term will also restrict the performance in equilirium models. We note that the symmetric constraints won't influence the final performance much as many works [29, 21] show.

A natural consideration arises as to whether we can utilize alternative kernel functions to extract input features for the equilibrium state. However, we find that other equilibrium models employing different kernels with inner products, like the polynomial kernel and sigmoid kernel, lead to similar structure to OptEqs with appropriate weight re-parameterization and lead to similar empirical results. We provide a detailed discussion of the related models in Appendix A.1. Thereby, we decide to use the Gaussian kernels and our new hidden optimization equation is formulated as follows:

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \min_{\mathbf{z}} \left[\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|_2^2 - \frac{1}{2\gamma} e^{-\gamma\|\mathbf{Ux}+\mathbf{b}-\mathbf{Wz}\|_2^2} \right], \tag{5}$$

where γ is the hyperparameter denoting the reciprocal of Gaussian kernels' variance for scaling. Calculating $\nabla G = \mathbf{0}$ for new G, we can get the Gaussian kernel inspired Equilibrium models (GEQ) as the following fixed-point equation:

$$\mathbf{z}^* = \sigma \left[e^{-\gamma\|\mathbf{Ux}+\mathbf{b}-\mathbf{Wz}^*\|_2^2} \mathbf{W}^\top (-\mathbf{Wz}^* + \mathbf{Ux} + \mathbf{b}) \right]. \tag{6}$$

Compared with linear kernels, Gaussian kernels can easily extract the non-linear relations from the input features and shows more stable and powerful performance in SVM and other machine learning methods [21, 41]. We also find that the formulation of our GEQ is similar to adding a new attention module to the original equilibrium models. Therefore, our GEQ is supposed to enjoy more representative abilities than the original OptEqs. In the following parts of this section, we will analyze the theoretical advantages of our GEQ against the vanilla OptEqs. And we also empirically evaluate GEQ's performance in the following sections.

3.2 GEQ equals to the OptEqs with infinite width

Like other Gaussian related models, our GEQ model can also be regarded as computing similarities by mapping them to an infinite-dimensional space. This allows GEQ to extract input features at the infinite-dimensional level, enabling the capture of non-linear dependencies in the input space. Essentially, our GEQ can be seen as a specialized version of OptEqs operating within the infinite-dimensional space after mapping input features $\mathbf{x}$ and output embedding $\mathbf{z}$ to this expanded domain.

Proposition 1. *The output of our GEQ (Eqn (6)) is the same as a special OptEqs' output whose hidden optimization problem is defined as follows:*

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \min_{\mathbf{z}} \left[\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|_2^2 - \lambda \left\langle \Phi_{\mathbf{U}}(\mathbf{x} + \mathbf{U}^{-1}\mathbf{b}), \Phi_{\mathbf{W}}(\mathbf{z}) \right\rangle \right], \tag{7}$$

where f is the positive indicator function and $(1+\partial f)^{-1}$ is the ReLU activation function, $\lambda = e^{-\gamma\|\mathbf{Ux}+\mathbf{b}\|_2^2} e^{-\gamma\|\mathbf{Wz}\|_2}$, and $\Phi_{\mathbf{W}}(\mathbf{z}) = [\mathbf{1}, \sqrt{\mathbf{2\gamma}}\mathbf{\Phi}_{\mathbf{W}}^{(\mathbf{1})}(\mathbf{z}), ..., \sqrt{(2\gamma)^i/i!}\Phi_{\mathbf{W}}^{(i)}(\mathbf{z}), ...] \in \mathbb{R}^{1\times\infty}$ which maps the inputs to the infinite-dimensional space with $\Phi_{\mathbf{W}}^{(i)} : \mathbb{R}^n \to \mathbb{R}^{in^i}$ defined as follows:

$$\Phi_{\mathbf{W}}^{(i)} = \left[\underbrace{\overbrace{(Wx)_0(Wx)_0...(Wx)_0}^{i}, \overbrace{(Wx)_0(Wx)_0...(Wx)_1}^{i}, ..., \overbrace{(Wx)_j(Wx)_k...(Wx)_m}^{i}, ...}_{in^i} \right], \tag{8}$$

where $(Wx)_j$ denotes the j-th element of vector $\mathbf{Wx}$.

Based on the analysis provided above, it becomes evident that the hidden optimization problem of our GEQ exhibits a similar formulation to a specific OptEqs, whose inputs $\mathbf{x}$ and outputs $\mathbf{z}$ are mapped to an infinite-dimensional space using the weight-tied infinite wide mapping $\Phi_{\mathbf{W}}$ and $\Phi_{\mathbf{U}}$. Given that both GEQ and OptEqs are derived from their respective hidden optimization problems, the equivalence in these problems implies the existence of the same equilibrium states for both models. Consequently, our GEQ can be considered an extension of the "infinite-depth" OptEqs to the "infinite-width" domain. Since wider neural networks are generally expected to perform better on classification tasks, we can infer that our GEQ outperforms vanilla equilibrium models like OptEqs. We further support this claim with theoretical analysis illustrated in the subsequent sections.

3.3 GEQ enjoys tighter generalization bound

Apart from the above empirical intuition, we are going to prove our GEQ's generalization advantages over OptEqs using the generalization bound under the PAC-Bayesian framework [33]. For convenience, we use $f_{geq}(\mathbf{x})$ denotes the equilibrium state $\mathbf{z}^*$ for input $\mathbf{x}$. Then we use the expected margin loss $\mathcal{L}_\eta(f_{geq}^c)$ at margin η of our GEQ on the data distribution $\mathcal{D}$ for classification, which is defined as follows,

$$\mathcal{L}_\eta(f_{geq}^c) = \mathbb{P}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[f_{geq}^c(\mathbf{x})_y \leq \eta + \max_{j\neq y} f_{geq}^c(\mathbf{x})_j\right], \tag{9}$$

where $f_{geq}^c(\mathbf{x}) = \mathbf{W}_c f_{geq}(\mathbf{x}) + \mathbf{b}_c$ stands for GEQ's final prediction at input $\mathbf{x}$ with learnable parameters $\mathbf{W}_c$ and $\mathbf{b}_c$, and the index j, y here denote the prediction score for certain class. Then we can analyze the generalization bound for our GEQ following the former work's settings [35].

Proposition 2. *If input $\|\mathbf{x}\|_2$ is bounded by B, $\mu := \max\{\|\mathbf{U}\|_2, \|\mathbf{W}\|_2, \|\mathbf{W}_c\|_2, \|\mathbf{b}\|_2\} < 1$, then we have following results for GEQ and OptEqs with ReLU activations. For $\forall\delta, \eta > 0$, with probability at least $1-\delta$ over the training set of size M, we have:*

$$\begin{aligned}\mathcal{L}_0(f_{geq}^c) \leq \hat{\mathcal{L}}_\eta(f_{geq}^c) + \sqrt{\frac{16h\ln(24h)\left[\beta_{\max}\mu^4 B + (2\mu\beta_{\max}+1)(1-\beta_{\max}m)\mu B + (1-\beta_{max}m)^2\right]^2 \mathcal{B}_W}{\eta^2(1-\beta_{\max}m)^4 M} + \frac{\ln(\frac{M\sqrt{M}}{\delta})}{M}}, \\ \mathcal{L}_0(f_{opteq}^c) \leq \hat{\mathcal{L}}_\eta(f_{opteq}^c) + \sqrt{\frac{16h\ln(24h)\left[\mu^3 B + (1-m)\mu B + (1-m)^2\right]^2 \mathcal{B}_W}{\eta^2(1-m)^4 M} + \frac{\ln(\frac{M\sqrt{M}}{\delta})}{M}},\end{aligned} \tag{10}$$

where $\hat{\mathcal{L}}_\eta(f_{geq}^c)$ denotes the empirical margin loss on the training set, the maximum scaling number is defined by $\beta_{max} := \max_{\mathbf{x}\in\mathcal{D}} e^{-\gamma\|\mathbf{U}\mathbf{x}+\mathbf{b}-\mathbf{W}\mathbf{z}\|_2^2}$, $\mathcal{B}_W := \|\mathbf{W}^\top\mathbf{W}\|_F^2 + \|\mathbf{U}\|_F^2 + \|\mathbf{b}\|_2^2 + \|\mathbf{W}_c\|_F^2 + \|\mathbf{b}_c\|_2^2$, and $m = \|\mathbf{W}^\top\mathbf{W}\|_2$ is less than 1 to ensure the convergence of equilibrium models.

Remark 1. *If $\beta_{\max} < 0.8$ and $\mu, m > 0.9$, we can get $\frac{\beta_{\max}\mu}{1-\beta_{\max}m} < \frac{1}{1-m}$ and $\frac{2\mu\beta_{\max}+1}{1-\beta_{\max}m} \leq \frac{1}{1-m}$. In the meanwhile, our GEQ's generalization bound is tighter than the original OptEq.*

In practical experiments, we find that the above conditions for $\beta_{\max}$ and μ, m are satisfied in most cases. Thereby, our GEQ's generalization bound is tighter than the original OptEqs in practical. Therefore, our GEQ can show better classification performance on the test set.

3.4 GEQ enjoys More Stable Performance

Apart from better performance, Gaussian kernel stands out as one of the most extensively employed kernels in machine learning tasks owing to its stability across various input scenarios. Motivated by this, we aim to investigate whether incorporating Gaussian kernels into our equilibrium models can enhance the model's stability across diverse inputs. Firstly, we are going to estimate output changes with respect to the input perturbations.

Proposition 3. *If norms for the inputs and outputs are bounded by B, the spectral norm for the weight parameter $\mathbf{W}, \mathbf{U}$ of equilibrium models with ReLU activation are bounded by $\mu < 1$ to ensure convergence, then we have the conclusions as below:*

$$\|f_{geq}(\mathbf{x}_1) - f_{geq}(\mathbf{x}_1)\|_2 \leq L_{geq}\|\mathbf{x}_1 - \mathbf{x}_2\|_2 = \frac{\beta_{\max}\mu^2 + \sqrt{\gamma}B\mu^3}{1-\beta_{\max}\mu^2 - \sqrt{\gamma}B\mu^3}\|\mathbf{x}_1 - \mathbf{x}_2\|_2, \tag{11}$$

$$\|f_{opteq}(\mathbf{x}_1) - f_{opteq}(\mathbf{x}_2)\|_2 \leq L_{opteq}\|\mathbf{x}_1 - \mathbf{x}_2\|_2 = \frac{\mu}{1-\mu^2}\|\mathbf{x}_1 - \mathbf{x}_2\|_2, \tag{12}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are input samples, $f_{geq}(\mathbf{x}_.)$ and $f_{opteq}(\mathbf{x}_.)$ denotes the equilibrium states for GEQ and OptEqs given input $\mathbf{x}_.$, and $\beta_{max} := \max_{\mathbf{x}\in\mathcal{D}} e^{-\gamma\|\mathbf{U}\mathbf{x}-\mathbf{W}\mathbf{z}\|_2^2} < 1$.

Remark 2. *If $\beta_{\max} < 0.8$, $B < 1$, and $\sqrt{\gamma} < 0.2$, then $L_{geq} < L_{opteq}$.*

In practical experiments, we choose different γ to reach the above condition for $\beta_{\max}$ and the condition for input B can also be achieved by normalization layers. Therefore, we can conclude that our GEQ's outputs are more stable under perturbations.

Besides having stable outputs under perturbations, a stable model should also show large output differences for different classes to make classification easier. However, the above Lipschitz term can not constrain outputs' similarity when samples are far apart, then we need a new metric for analysis. In line with previous works [18, 31, 10], we assume all weight parameters go to infinite dimensions and analyze the expected output similarity κ for a model f for inputs $\mathbf{x}_1$ and $\mathbf{x}_2$ defined below:

$$\kappa(\mathbf{x}_1,\mathbf{x}_2) = \mathbb{E}\left[f(\mathbf{x}_1)^\top f(\mathbf{x}_2)\right] = \int_{\mathbb{R}} f_{\mathbf{u}}(\mathbf{x}_1)^\top f_{\mathbf{u}}(\mathbf{x}_2) p(\mathbf{u}) d\mathbf{u}, \tag{13}$$

with $p(\mathbf{u})$ is the distribution of weight $\mathbf{U}$'s vectorization. If κ is smaller for samples $\mathbf{x}_1$ and $\mathbf{x}_2$ when they belong to different classes, which means they are far away, then the classifier can easily classify these two samples with different labels. The margin for the classification will also be large and easy for the classification of difficult samples. The κ's upper bound for GEQ and OptEqs are listed below:

Proposition 4. *If norms for the inputs and outputs are bounded by B, the spectral norm for the weight parameter $\mathbf{W}$ of equilibrium models with ReLU activation are bounded by $\mu < 1$ to ensure the convergence, and each row in $\mathbf{U}$ obeys the spherical Gaussian distributions $\mathcal{N}(0, \mathbb{E}[U_i^2]\mathbf{I})$. Then we have the following conclusions for the expectation of the output similarity for GEQ and OptEqs with respect to input $\mathbf{x}_1, \mathbf{x}_2$ as follows,*

$$\kappa_{geq}(\mathbf{x}_1,\mathbf{x}_2) \le \overline{\kappa}_{geq} = \frac{\mu^2 D e^{-\frac{\gamma}{4}(\sigma_{\min}(\mathbf{U})^2\|\mathbf{x}_1-\mathbf{x}_2\|_2^2)}\mathbb{E}[U_i^2]\|\mathbf{x}_1\|_2\|\mathbf{x}_2\|_2\left(\sin\theta_0+(\pi-\theta_0)\cos\theta_0\right)}{2\pi(1-\beta_{\max}\mu^2)^2}, \tag{14}$$

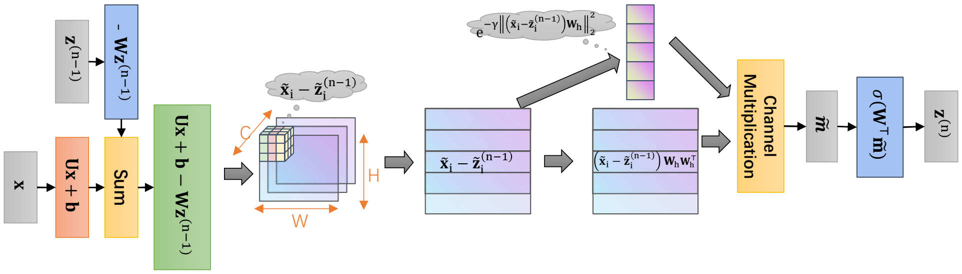
$$\kappa_{opteq}(\mathbf{x}_1,\mathbf{x}_2) \le \overline{\kappa}_{opteq} = \frac{\mathbb{E}[U_i^2]\|\mathbf{x}_1\|_2\|\mathbf{x}_2\|_2\left(\sin\theta_0+(\pi-\theta_0)\cos\theta_0\right)}{2\pi(1-\mu^2)^2}, \tag{15}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are input samples, $D = e^{\gamma B\|\mathbf{W}\|_2^2}$, $\beta_{max} := \max_{\mathbf{x}\in\mathcal{D}} e^{-\gamma\|\mathbf{Ux}-\mathbf{Wz}\|_2^2}$, $\sigma_{\min}(\mathbf{U})$ is $\mathbf{U}$'s minimal singular term, and $\theta_0 = \cos^{-1}(\frac{\langle\mathbf{x}_1,\mathbf{x}_2\rangle}{\|\mathbf{x}_1\|\|\mathbf{x}_2\|})$ is the angle between the samples.

Remark 3. *If $\|\mathbf{x}_1-\mathbf{x}_2\|_2 \ge 2\sqrt{-log(1/D)}/\sigma_{\min}(\mathbf{U})$, then $\overline{\kappa}_{geq} \le \overline{\kappa}_{geq}$.*

Based on the aforementioned analysis, it is evident that our GEQ exhibits a smaller output similarity for dissimilar samples. As a result, the predictions made by GEQ are primarily based on the most similar samples, enabling it to successfully classify challenging instances. This claim is further supported by the results obtained from our carefully designed experiments.

3.5 Patch Splitting in GEQ



$\mathbf{z}^{(n-1)}$
$-\mathbf{Wz}^{(n-1)}$
$\mathbf{x}$
$\mathbf{Ux+b}$
Sum
$\mathbf{Ux+b-Wz}^{(n-1)}$
$\tilde{\mathbf{x}}_i - \tilde{\mathbf{z}}_i^{(n-1)}$
C
H
W
$\tilde{\mathbf{x}}_i - \tilde{\mathbf{z}}_i^{(n-1)}$
$e^{-\gamma\|(\tilde{\mathbf{x}}_i-\tilde{\mathbf{z}}_i^{(n-1)})\mathbf{W}_h\|_2^2}$
$(\tilde{\mathbf{x}}_i - \tilde{\mathbf{z}}_i^{(n-1)})\mathbf{W}_h\mathbf{W}_h^\top$
Channel Multiplication
$\tilde{\mathbf{m}}$
$\sigma(\mathbf{W}^\top\tilde{\mathbf{m}})$
$\mathbf{z}^{(n)}$

Figure 1: The sketch map of one layer GEQ's n-th fixed point iteration. $\mathbf{x}$ is the input and $\mathbf{z}^{(n-1)}$, $\mathbf{z}^{(n)}$ are the output of $(n-1)$-th, n-th iteration, and $\tilde{\mathbf{m}}$ is a middle state.

Since different parts of images have different impacts on the image classification, calculating the whole similarity using Gaussian kernels for GEQ is not enough. Inspired by former works [44], we also split the feature map $\mathbf{Ux}$ into patches, and then our optimization problem becomes:

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \min_{\mathbf{z}} \left[\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|_2^2 - \frac{1}{2\gamma}\sum_{i=1}^{N} e^{-\gamma\|(\tilde{\mathbf{x}}_i-\tilde{\mathbf{z}}_i)\mathbf{W}_h\|_2^2}\right], \tag{16}$$

where $\tilde{\mathbf{x}}_i \in \mathbb{R}^{c_s p^2}$ is the i-th patch of $\mathbf{Ux}+\mathbf{b}$ while $\tilde{\mathbf{z}}_i \in \mathbb{R}^{c_s p^2}$ is the i-th patch of $\mathbf{Wz}$ and $\mathbf{W}_h \in \mathbb{R}^{c_s p^2 \times c_{hid}}$ is a linear layer to project patches with different size to the constant dimension. c_s denotes the channel splitting number, p denotes the patch size, and c_{hid} denotes the hidden dimension of patches after projection. We note that the patch-splitting approach is a GEQ's unique feature, as incorporating this technique makes no difference in OptEqs due to its linear kernel. Figure 1 provides a sketch for GEQ's i-th fixed-point iteration. From the figure, it is evident that our GEQ can be viewed as a special OptEqs with additional attention mechanisms to capture the most important regions. Thereby, it can achieve enhanced performance. For a more detailed understanding of the forward procedure in our GEQ, please refer to Appendix A.2.

4 Empirical Results

4.1 Experiment Settings

In our experiments, we employed parallel GEQs with different input scales like MOptEqs and averaged the output of each branch after average pooling or nearest up-sampling to fuse the branches. We use weight normalizations to ensure the convergence as MOptEqs and MDEQ, and set γ to $0.2/M$, where M is the minimum $\|\tilde{\mathbf{x}}. - \tilde{\mathbf{z}}.\mathbf{W}_h\|_2^2$ among all patches. For the equilibrium calculation, we used the Anderson algorithm in the forward procedure, similar to other implicit models [25], and applied Phantom gradients [14] for back-propagation. All models were trained using SGD with a step learning rate schedule. We implemented our experiments on the PyTorch platform [36] using RTX-3090. Further details can be found in the Appendix A.7. To compare the performance of our GEQ, we used MOptEqs and MDEQ as benchmark implicit models, which have demonstrated superior performance over OptEqs on image classification tasks. Additionally, we used ResNet-18 and ResNet-50 as benchmark explicit models for comparison.

4.2 Results for Image Classification

Firstly, we finish the experiments on CIFAR-10 and CIFAR-100. They are widely used datasets for image classification on small images. In the experiment, we parallel 6 branches GEQ with the input scale is $32, 16, 8, 8, 4, 4$ and MOptEqs' architecture setting is also the same. The details can be found in the Appendix. As for the comparison, we also conduct experiments of the same training procedure for MDEQ, MOptEqs, and ResNet. The results are listed in Table 1.

	Model Size	Accuracy
ResNet-18	10M	$93.5 \pm 0.2\%$
ResNet-50	23M	95.2 ± 0.2
MDEQ	10M	$94.2 \pm 0.3\%$
MOptEqs	8M	$94.6 \pm 0.2\%$
GEQ	5M	$94.8 \pm 0.1\%$
GEQ	8M	$\mathbf{95.6 \pm 0.2}\%$

(a) CIFAR-10.

	Model Size	Accuracy
ResNet-18	10M	$74.5 \pm 0.2\%$
ResNet-50	23M	$77.9 \pm 0.1\%$
MDEQ	10M	$74.7 \pm 0.3\%$
MOptEqs	8M	$75.6 \pm 0.2\%$
GEQ	5M	$76.4 \pm 0.3\%$
GEQ	8M	$\mathbf{78.2 \pm 0.2}\%$

(b) CIFAR-100.

Table 1: The Empirical results for image classification on CIFAR-10 and CIFAR-100.

From the results, one can see that our GEQ enjoys clear advantages on CIFAR datasets, which demonstrates the powerful generalization ability of other models.

	Model Size	Accuracy
ResNet-18	11M	$92.3 \pm 0.1\%$
ResNet-50	23M	$93.0 \pm 0.2\%$
MDEQ	10M	$91.5 \pm 0.2\%$
MOptEqs	10M	$92.4 \pm 0.2\%$
GEQ	6M	$\mathbf{92.9 \pm 0.2}\%$
GEQ	13M	$\mathbf{93.2 \pm 0.1}\%$

(a) ImageNette.

	Model Size	Accuracy
ResNet-18	11M	$80.9 \pm 0.3\%$
ResNet-50	23M	$81.7 \pm 0.2\%$
MDEQ	10M	$81.3 \pm 0, 2\%$
MOptEqs	13M	$81.5 \pm 0.4\%$
GEQ	6M	$\mathbf{82.2 \pm 0.2}\%$
GEQ	13M	$\mathbf{83.9 \pm 0.3}\%$

(b) ImageNet-100.

Table 2: The Empirical results for image classification on ImageNette and ImageNet-100.

Besides small datasets, we also conducted experiments on large-scale image datasets, as presented in Table 2. The results clearly demonstrate the consistent superiority of our GEQ over other models, highlighting its clear advantages. Particularly noteworthy is our GEQ achieves a 2% improvement on ImageNet-100 agaginst deep model ResNet-50 while consuming approximately half the number of parameters, which emphasizes the effectiveness and efficiency of GEQ on large-scale inputs.

4.3 Validations on the models' stability

Evaluation on Unseen difficult samples. In order to assess the stability of our GEQ model on difficult examples, we conducted experiments using CIFAR-100 super-class classification. CIFAR-100 consists of 20 super classes, each containing five sub-classes [1]. We trained our GEQ and MOptEqs models to predict the super-classes using the first four sub-classes from each super-class for training. We evaluated the models using both the test set, which includes the first four sub-classes from each super-class (referred to as "Known Accuracy"), and a separate set of samples from unseen sub-classes (referred to as "Unknown Accuracy"). The classification of the unseen samples is more difficult as they are different from the training set. The results of our GEQ and MOptEqs models are presented in Table 3.

	Known Accuracy	Unknown Accuracy
MOptEqs	$80.1 \pm 0.3\%$	$77.4 \pm 0.5\%$
GEQ	$\mathbf{80.9} \pm \mathbf{0.2}\%$	$\mathbf{80.1} \pm \mathbf{0.6}\%$

Table 3: Empirical rsults on CIFAR-100's super-class classification.

The above table clearly demonstrates that our GEQ model surpasses MOptEqs in achieving superior performance on the challenging task at hand and demonstrates GEQ's stability. Such advantages can be attributed to the fact that GEQ exhibits smaller output similarities compared to OptEqs when input samples are far apart (e.g., samples from different classes). This characteristic can lead to larger margins between different classes, enabling the classifier to be more easily optimized during training. Consequently, our GEQ model excels in accurately classifying difficult unseen samples, further highlighting its stability and superiority over former equilibrium models.

Ablation Studies on corrupted datasets. Apart from difficult samples, we are going to compare the robustness of our GEQ, MOptEqs, and ResNet on the CIFAR-10 corruption dataset, which contains 19 common corruptions including image transformation, image blurring, weather, and digital noises on CIFAR's test datasets. The average results on 5 degrees CIFAR-10 corruption datasets are drawn in Figure 2.

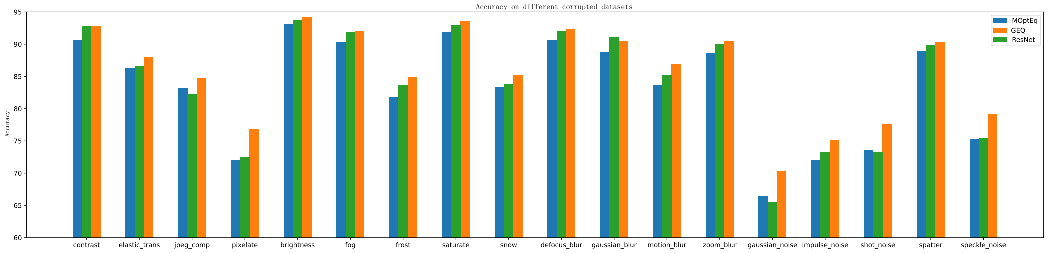


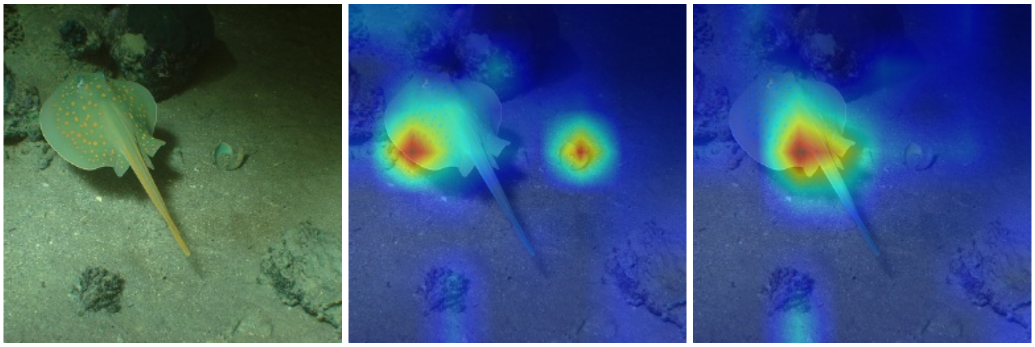
Figure 2: The results for different models under different corruptions.

From the result, one can see that our GEQ based on Gaussian kernels is more robust than MOptEqs and ResNet. Especially, our GEQ can show better performance against structured noise and some image transformation. The above results also demonstrate the stability of our GEQ structure.

4.4 Ablation Studies on Saliency Map

The saliency maps generated by GradCAM [42] offer valuable insights into the visual attention of both MOptEqs and GEQ models. These maps highlight the regions of the image that are crucial for the model's predictions. Figure 3 presents the saliency maps obtained for an image from the ImageNet dataset using both models. Upon observation, it becomes evident that GEQ exhibits a higher degree of focus on the significant regions directly associated with the predicted label "manta". In contrast, MOptEqs tends to allocate attention to unrelated regions such as the shells. This discrepancy indicates that the attention-like module induced by the Gaussian kernel in GEQ enhances the concentration of the model's attention, resulting in improved performance compared to MOptEqs.

[1] For example, super class "people" contains five sub classes:"baby","girl","man","man","woman"

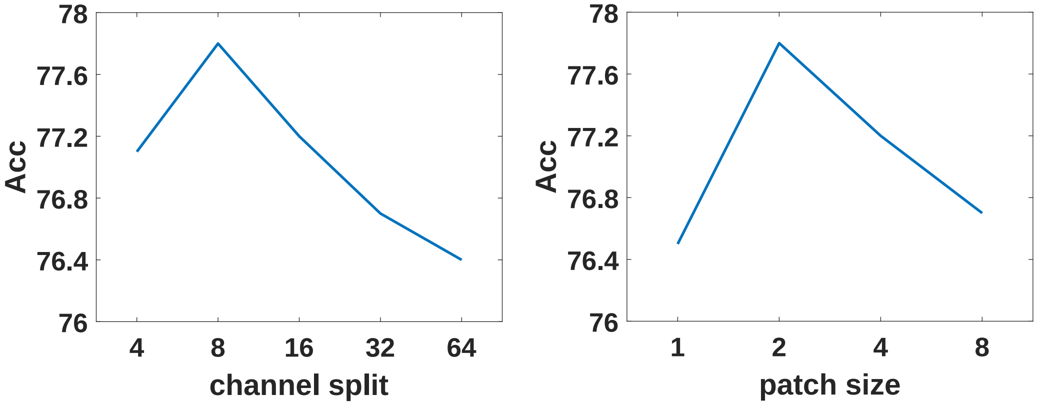


(a) Original Image. (b) MOptEqs's saliency map. (c) GEQ's saliency map.

Figure 3: Saliency map for GEQ and MOptEqs on the ImageNet image.

4.5 Ablation Studies on Patch splitting

The performance of our GEQ model is influenced by the channel splitting parameter and the patch size. Choosing large values for these parameters causes the kernel to focus mainly on global information while selecting small values makes the kernel concentrate on local features. To understand the impact of these choices on model performance, we conducted experiments, the results of which are presented in Figure 4. This figure illustrates the relationship between the channel splitting parameter, patch size, and the model's performance. By analyzing these results, we gain insights into the optimal values for these parameters that yield the best performance for our GEQ model.



(a) Influence of the channel splitting parameter. (b) Influence of the patch size parameter.

Figure 4: The influence on the patch size and the channel splitting parameter for our GEQ on CIFAR-100 datasets.

The accuracy trend depicted in the figure shows an initial increase followed by a decrease as the channel split and patch size increase. Based on these empirical results, we select a patch size of 2 and a channel split of 8 for both the CIFAR and ImageNet experiments. These parameter choices are made to optimize the performance of our models on the respective datasets.

5 Conclusions

In this paper, we introduce a novel optimization induced equilibrium model called GEQ, which utilizes Gaussian kernels in its optimization-induced framework. Our model incorporates a new attentive module that arises from its novel hidden optimization problem formulation. Notably, GEQ exhibits significantly improved performance in classification tasks, outperforming deep models as well. Moreover, GEQ can be interpreted as a weight-tied model with infinite width and depth, highlighting its expressive power. We also provide theoretical analysis demonstrating the superiority of our models in terms of generalization ability and stability compared to previous OptEqs. Empirical results further validate the effectiveness of our proposed approach.

References

[1] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017.

[2] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

[4] Shaojie Bai, Vladlen Koltun, and J Zico Kolter. Multiscale deep equilibrium models. *Advances in Neural Information Processing Systems*, 33:5238–5250, 2020.

[5] Minmin Chen, Jeffrey Pennington, and Samuel Schoenholz. Dynamical isometry and a mean field theory of rnns: Gating enables signal propagation in recurrent neural networks. In *International Conference on Machine Learning*, pages 873–882. PMLR, 2018.

[6] Qi Chen, Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Optimization-induced graph implicit nonlinear diffusion. In *International Conference on Machine Learning*, pages 3648–3661. PMLR, 2022.

[7] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6572–6583, 2018.

[8] KR1442 Chowdhary. Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649, 2020.

[9] Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in neural information processing systems*, 31:7178–7189, 2018.

[10] Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[11] Josip Djolonga and Andreas Krause. Differentiable learning of submodular models. *Advances in Neural Information Processing Systems*, 30:1013–1023, 2017.

[12] Zhili Feng and J Zico Kolter. On the neural tangent kernel of equilibrium models, 2021.

[13] Adrià Garriga-Alonso, Carl Edward Rasmussen, and Laurence Aitchison. Deep convolutional networks as shallow gaussian processes. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[14] Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. On training implicit models. *Advances in Neural Information Processing Systems*, 34:24247–24260, 2021.

[15] Laurent El Ghaoui, Fangda Gu, Bertrand Travacca, and Armin Askari. Implicit deep learning. *CoRR*, abs/1908.06315, 2019.

[16] Dar Gilboa, Bo Chang, Minmin Chen, Greg Yang, Samuel S Schoenholz, Ed H Chi, and Jeffrey Pennington. Dynamical isometry and a mean field theory of lstms and grus. *arXiv preprint arXiv:1901.08987*, 2019.

[17] Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks: A new hope. *CoRR*, abs/1909.04866, 2019.

[18] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[19] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. Mean-field behaviour of neural tangent kernel for deep neural networks. *arXiv preprint arXiv:1905.13654*, 2019.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Shell Xu Hu, Sergey Zagoruyko, and Nikos Komodakis. Exploring weight symmetry in deep neural networks. *Computer Vision and Image Understanding*, 187:102786, 2019.

[22] Arthur Jacot, Franck Gabriel, and Clément Hongler. Freeze and chaos for dnns: an NTK view of batch normalization, checkerboard and boundary effects. *CoRR*, abs/1907.05715, 2019.

[23] Nikhil Ketkar and Nikhil Ketkar. Stochastic gradient descent. *Deep learning with Python: A hands-on introduction*, pages 113–132, 2017.

[24] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[25] Mingjie Li, Yisen Wang, Xingyu Xie, and Zhouchen Lin. Optimization inspired multi-branch equilibrium models. In *International Conference on Learning Representations*, 2021.

[26] Guangcan Liu, Shiyu Chang, and Yi Ma. Blind image deblurring using spectral properties of convolution operators. *IEEE Transactions on image processing*, 23(12):5047–5056, 2014.

[27] Guangcan Liu and Ping Li. Low-rank matrix completion in the presence of high coherence. *IEEE Transactions on Signal Processing*, 64(21):5623–5633, 2016.

[28] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[29] Juncheng Liu, Kenji Kawaguchi, Bryan Hooi, Yiwei Wang, and Xiaokui Xiao. Eignn: Efficient infinite-depth graph neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[30] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696, 2009.

[31] Lassi Meronen, Christabella Irwanto, and Arno Solin. Stationary activations for uncertainty calibration in deep learning. *Advances in Neural Information Processing Systems*, 33:2338–2350, 2020.

[32] Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer, 1996.

[33] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *CoRR*, abs/1707.09564, 2017.

[34] Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Jiri Hron, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[35] Chirag Pabbaraju, Ezra Winston, and J Zico Kolter. Estimating lipschitz constants of monotone deep equilibrium models. In *International Conference on Learning Representations*, 2021.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[37] Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming Lin. Scalable differentiable physics for learning and control. In *International Conference on Machine Learning*, pages 7847–7856. PMLR, 2020.

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[39] Max Revay, Ruigang Wang, and Ian R Manchester. Lipschitz bounded equilibrium networks. *arXiv preprint arXiv:2010.01732*, 2020.

[40] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[41] Bernhard Scholkopf, Kah-Kay Sung, Christopher JC Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE transactions on Signal Processing*, 45(11):2758–2765, 1997.

[42] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[43] Ivana Tošić and Pascal Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, 2011.

[44] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

[45] Russell Tsuchida, Fred Roosta, and Marcus Gallagher. Invariance of weight distributions in rectified mlps. In *International Conference on Machine Learning*, pages 4995–5004. PMLR, 2018.

[46] Russell Tsuchida, Suk Yee Yong, Mohammad Ali Armin, Lars Petersson, and Cheng Soon Ong. Declarative nets that are equilibrium models. In *International Conference on Learning Representations*, 2022.

[47] Po-Wei Wang, Priya Donti, Bryan Wilder, and Zico Kolter. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, pages 6545–6554. PMLR, 2019.

[48] Ezra Winston and J. Zico Kolter. Monotone operator equilibrium networks. *CoRR*, abs/2006.08591, 2020.

[49] Xingyu Xie, Jianlong Wu, Guangcan Liu, Zhisheng Zhong, and Zhouchen Lin. Differentiable linearized admm. In *International Conference on Machine Learning*, pages 6902–6911. PMLR, 2019.

[50] Xingyu Xie, Jianlong Wu, Guangcan Liu, Zhisheng Zhong, and Zhouchen Lin. Optimization induced deep equilibrium networks. *arXiv preprint arXiv:2105.13228*, 2021.

[51] Chong You, Daniel Robinson, and René Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3918–3927, 2016.

[52] Hongyang Zhang, Zhouchen Lin, Chao Zhang, and Edward Y Chang. Exact recoverability of robust pca via outlier pursuit with tight recovery bounds. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[53] Xiao Zhang, Lingxiao Wang, Yaodong Yu, and Quanquan Gu. A primal-dual analysis of global optimality in nonconvex low-rank matrix recovery. In *International conference on machine learning*, pages 5862–5871. PMLR, 2018.

[54] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

A Appendix

A.1 Comparison with other kernel functions

Firstly, we introduce different commonly used kernels, such as polynomial, sigmoid, and Gaussian kernels, to reformulate the hidden optimization problem for equilibrium models. Table A.1 illustrates the equilibrium models induced by these kernels. It can be observed that equilibrium models with polynomial and sigmoid kernels also incorporate new attentive modules. However, their attentive kernels only constrain the input features $\mathbf{Ux} + \mathbf{b}$ and do not directly affect the activation of the output $\mathbf{z}^*$. As a result, their performance may be inferior to our GEQ model. To validate these claims, we evaluate the performance of different models on the CIFAR-100 dataset. Since the fusion module is the primary difference between MOptEqs and OptEqs, we can easily modify the structure of MOptEqs to include different kernel-induced attentive modules using the equations in Table A.1, resulting in MOptEqs (Polynomial) and MOptEqs (Sigmoid). The results are presented in Table A.1, which clearly demonstrates the superior performance of our GEQ model. For a more in-depth analysis of GEQ, we refer readers to the main paper.

Kernel	Hidden Optimization Problem	Equilibrium Model
Linear	$\min_{\mathbf{z}} \left[\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|_2^2 - \langle \mathbf{Ux}+\mathbf{b}, \mathbf{z}\rangle - \frac{1}{2}\|\mathbf{Wz}\|_2^2\right]$	$\mathbf{z}^* = \sigma\left(\mathbf{W}^\top\mathbf{Wz}^* + \mathbf{Ux} + \mathbf{b}\right)$
Polynomial	$\min_{\mathbf{z}} \left[\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|_2^2 - \left(\langle \mathbf{Ux}+\mathbf{b}, \mathbf{z}\rangle\right)^d - \frac{1}{2}\|\mathbf{Wz}\|_2^2\right]$	$\mathbf{z}^* = \sigma\left(\mathbf{W}^\top\mathbf{Wz}^* + d\left(\langle \mathbf{Ux}+\mathbf{b}, \mathbf{z}\rangle\right)^{d-1}(\mathbf{Ux} + \mathbf{b})\right)$
Sigmoid	$\min_{\mathbf{z}} \left[\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|_2^2 - \tanh\left(\langle \mathbf{Ux}+\mathbf{b}, \mathbf{z}\rangle\right) - \frac{1}{2}\|\mathbf{Wz}\|_2^2\right]$	$\mathbf{z}^* = \sigma\left(\mathbf{W}^\top\mathbf{Wz}^* + \left(1 - \tanh^2\left(\langle \mathbf{Ux}+\mathbf{b}, \mathbf{z}\rangle\right)\right)(\mathbf{Ux} + \mathbf{b})\right)$
Gaussian	$\min_{\mathbf{z}} \left[\mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|_2^2 - \frac{1}{2\gamma}e^{-\gamma\|\mathbf{Ux}+\mathbf{b}-\mathbf{Wz}\|_2^2}\right]$	$\mathbf{z}^* = \sigma\left[e^{-\gamma\|\mathbf{Ux}+\mathbf{b}-\mathbf{Wz}^*\|_2^2}\mathbf{W}^\top(-\mathbf{Wz}^* + \mathbf{Ux} + \mathbf{b})\right]$

Table 4: The hidden optimization problems and their related equilibrium models. $d > 1$ is an integer denoting the polynomial order.

	Model Size	Accuracy
MOptEqs	8M	$75.6 \pm 0.2\%$
MOptEqs (Polynomial)	8M	$75.1 \pm 0.4\%$
MOptEqs (Sigmoid)	8M	$76.1 \pm 0.3\%$
GEQ	8M	$\mathbf{78.2 \pm 0.2\%}$

Table 5: Comparison of equilibrium models with different kernel functions on CIFAR-100.

A.2 Forward Procedure for GEQ

The pseudo-code for our GEQ is listed in Algorithm 1.

Algorithm 1: Calculating one layer GEQ.

```
Require: initial state $\mathbf{z}^{(0)}$, weight parameter $\mathbf{W}$, $\mathbf{Ux} \in \mathbb{R}^{chw}$, channel split $c_s$, patch size $p$, hidden layer
  $\mathbf{W}_h \in \mathbb{R}^{c_s p^2 \times 32}$
Ensure: Get the output $\mathbf{z}^*$ of $i$-th fixed point iteration.
  def $g(\mathbf{z}^{(i)}; \mathbf{x}, \mathbf{U}, \mathbf{W}, \mathbf{b})$:
      Rearrage $\mathbf{Wz}^{(i)} \to \tilde{\mathbf{z}}$, $\mathbf{Ux} + \mathbf{b} \to \tilde{\mathbf{x}} \in \mathbb{R}^{\frac{chw}{c_s p^2} \times (c_s p^2)}$
      $\tilde{\mathbf{m}} = diag\left(e^{-\gamma\|(\tilde{\mathbf{x}}_{\cdot} - \tilde{\mathbf{z}}_{\cdot})\mathbf{W}_h\|_2^2}\right)(\tilde{\mathbf{x}} - \tilde{\mathbf{z}})\mathbf{W}_h\mathbf{W}_h^\top$
      Rearrage $\tilde{\mathbf{m}} \to \mathbf{m} \in \mathbb{R}^{c \times h \times w}$
      return $\mathbf{z}^{(i+1)} = \sigma\left(\mathbf{W}^\top\mathbf{m}\right)$
  with torch.no_grad():
      Use anderson algorithm to solve $\mathbf{z}^* = g(\mathbf{z}^*; \mathbf{x}, \mathbf{U}, \mathbf{W}, \mathbf{b})$
  # calculate gradient via phantom gradient:
  for i in range(5) do
    $\mathbf{z}^* = 0.2 \times \mathbf{z}^* + 0.8 \times g(\mathbf{z}^*; \mathbf{x}, \mathbf{U}, \mathbf{W}, \mathbf{b})$.
  end for
  return $\mathbf{z}^*$
```

A.3 Proofs for proposition 1

Proposition 1. *The outputs of GEQ with Gaussian kernels (Eqn (6)) is the same as Optimized induced Equilibrium Models' output whose output is the optimal solution of the hidden optimization problems:*

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|^2 - \lambda\left\langle \Phi_{\mathbf{U}}(\mathbf{x}+\mathbf{U}^{-1}\mathbf{b}), \Phi_{\mathbf{W}}(\mathbf{z})\right\rangle \tag{17}$$

where $\Phi_{\mathbf{W}}(\mathbf{z}) = \left[\mathbf{1}, \sqrt{\mathbf{2}\gamma}\mathbf{\Phi}_{\mathbf{W}}^{(\mathbf{1})}(\mathbf{z}), ..., \sqrt{\frac{(2\gamma)^i}{i}}\Phi_{\mathbf{W}}^{(i)}(\mathbf{z}), ...\right] \in \mathbb{R}^{1\times\infty}$ *which projects the features to the infinite-dimensional space. And* $\Phi_{\mathbf{W}}^{(i)} : \mathbb{R}^n \to \mathbb{R}^{in^i}$ *is the k-tuple permutation with repetitions formulated as follows:*

$$\Phi_{\mathbf{W}}^{(i)} = \left[\underbrace{\overbrace{(Wx)_1(Wx)_1...(Wx)_1}^{i}, \overbrace{(Wx)_1(Wx)_1...(Wx)_2}^{i}, ..., \overbrace{(Wx)_j(Wx)_k...(Wx)_m}^{i}, ...}_{in^i}\right] \tag{18}$$

where $(Wx)_j$ *denotes the* j*-th element of vector* $\mathbf{Wx}$*. Then the Gaussian kernel can also be regarded as calculating the input features after the weight-tied infinite wide projection* $\Phi_{\mathbf{W}}$ *and* $\Phi_{\mathbf{U}}$*.*

Proof. We can formulate the OptEqs' hidden optimization problem with Gaussian kernels as below:

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|^2 - \frac{1}{2\gamma}e^{-\gamma\|\mathbf{Ux}+\mathbf{b}-\mathbf{Wz}\|_2^2}, \tag{19}$$

For $e^{-\gamma\|(\mathbf{Ux}+\mathbf{b}-\mathbf{Wz})\|_2^2}$, we have

$$\begin{aligned} e^{-\gamma\|\mathbf{Ux}+\mathbf{b}-\mathbf{Wz}\|_2^2} &= e^{-\gamma\|\mathbf{Ux}+\mathbf{b}\|_2^2-\gamma\|\mathbf{Wz}\|_2^2+2\gamma\langle\mathbf{Ux}+\mathbf{b},\mathbf{Wz}\rangle}, \\ &= e^{-\gamma\|\mathbf{Ux}+\mathbf{b}\|_2^2-\gamma\|\mathbf{Wz}\|_2}e^{2\gamma\langle\mathbf{Ux}+\mathbf{b},\mathbf{Wz}\rangle}, \end{aligned} \tag{20}$$

letting $\lambda = e^{-\gamma\|\mathbf{Ux}+\mathbf{b}\|_2^2}e^{-\gamma\|\mathbf{Wz}\|_2}$ and we do the Taylor expansion for $e^{\langle\mathbf{Ux}+\mathbf{b},\mathbf{Wz}\rangle}$, we have:

$$e^{-\gamma\|\mathbf{Ux}+\mathbf{b}-\mathbf{Wz}\|_2^2} = \lambda\sum_{i=0}^{\infty}\frac{\left(\langle\sqrt{2\gamma}(\mathbf{Ux}+\mathbf{b}), \sqrt{2\gamma}(\mathbf{Wz})\rangle\right)^i}{i!}. \tag{21}$$

For any i we have from the permutation theory:

$$\frac{\left(\langle\sqrt{2\gamma}(\mathbf{Ux}+\mathbf{b}), \sqrt{2\gamma}(\mathbf{Wz})\rangle\right)^i}{i!} = \lambda\left\langle\Phi_{\mathbf{U}}^{(i)}(\mathbf{x}+\mathbf{U}^{-1}\mathbf{b}), \Phi_{\mathbf{W}}^{(i)}(\mathbf{z})\right\rangle, \tag{22}$$

where $\Phi_{\mathbf{W}}^{(i)}(\mathbf{x})$ is the i-tuple permutation with the repetition for given $(Wx)_1, (Wx)_2, ..., (Wx)_n$. Each element of $\Phi_{\mathbf{W}}^{(i)}(\mathbf{x})$ is one possible permutation. Since there are n^i tuples, then $\Phi_{\mathbf{W}}^{(i)}$ can project the input features to in^i space as follows,

$$\Phi_{\mathbf{W}}^{(i)} = \left[\underbrace{\overbrace{(Wx)_1(Wx)_1...(Wx)_1}^{i}, \overbrace{(Wx)_1(Wx)_1...(Wx)_2}^{i}, ..., \overbrace{(Wx)_j(Wx)_k...(Wx)_m}^{i}, ...}_{in^i}\right]. \tag{23}$$

Thereby, the hidden optimization problem for our GEQ can be reformulate as,

$$\min_{\mathbf{z}} G(\mathbf{z};\mathbf{x}) = \mathbf{1}^\top f(\mathbf{z}) + \frac{1}{2}\|\mathbf{z}\|^2 - \lambda\left\langle \Phi_{\mathbf{U}}(\mathbf{x}+\mathbf{U}^{-1}\mathbf{b}), \Phi_{\mathbf{W}}(\mathbf{z})\right\rangle \tag{24}$$

Then our conclusion are proved. □

A.4 Proofs for proposition 2

Proposition 2. *If input $\|\mathbf{x}\|_2$ is bounded by B, $\mu := \max\{\|\mathbf{U}\|_2, \|\mathbf{W}\|_2, \|\mathbf{W}_c\|_2, \|\mathbf{b}\|_2\} < 1$, then we have following results for GEQ and OptEqs with ReLU activations. For $\forall \delta, \eta > 0$, with probability at least $1 - \delta$ over the training set of size M, we have:*

$$\begin{aligned}\mathcal{L}_0(f^c_{geq}) \leq \hat{\mathcal{L}}_\eta(f^c_{geq}) + \sqrt{\frac{16h\mathrm{ln}(24h)\left[\beta_{\max}\mu^4 B + (2\mu\beta_{\max}+1)(1-\beta_{\max}m)\mu B + (1-\beta_{max}m)^2\right]^2 \mathcal{B}_W}{\eta^2(1-\beta_{\max}m)^4 M} + \frac{\ln(\frac{M\sqrt{M}}{\delta})}{M}},\\ \mathcal{L}_0(f^c_{opteq}) \leq \hat{\mathcal{L}}_\eta(f^c_{opteq}) + \sqrt{\frac{16h\mathrm{ln}(24h)\left[\mu^3 B + (1-m)\mu B + (1-m)^2\right]^2 \mathcal{B}_W}{\eta^2(1-m)^4 M} + \frac{\ln(\frac{M\sqrt{M}}{\delta})}{M}},\end{aligned} \tag{25}$$

where $\hat{\mathcal{L}}_\eta(f^c_{geq})$ denotes the empirical margin loss on the training set, the maximum scaling number is defined by $\beta_{max} := \max\limits_{\mathbf{x}\in\mathcal{D}} e^{-\gamma\|\mathbf{Ux}+\mathbf{b}-\mathbf{Wz}\|_2^2}$, $\mathcal{B}_W := \|\mathbf{W}^\top\mathbf{W}\|_F^2 + \|\mathbf{U}\|_F^2 + \|\mathbf{b}\|_2^2 + \|\mathbf{W}_c\|_F^2 + \|\mathbf{b}_c\|_2^2$, and $m = \|\mathbf{W}^\top\mathbf{W}\|_2$ is less than 1 to ensure the convergence of equilibrium models.

Before the proof our bounds, we need to introduce a lemma in former work [35] for the perturbation bound for GEQs and reformulated OptEqs as follows.

Lemma 1. *Let $\|\mathbf{W}\|_2 \leq m$ and $\|\overline{\mathbf{W}}\|_2 \leq m$. Then change in the output of the DEqs $\mathbf{z} = \sigma(\mathbf{Wz} + \mathbf{Ux} + \mathbf{b})$ on perturbation the weights and biases from $\mathbf{W}, \mathbf{U}, \mathbf{b}$ to $\overline{\mathbf{W}}, \overline{\mathbf{U}}, \overline{\mathbf{b}}$ is bounded as follows:*

$$\begin{aligned}\left\|f(\overline{\mathbf{W}}\mathbf{z} + \overline{\mathbf{U}}\mathbf{x} + \overline{\mathbf{b}}) - f(\mathbf{Wz} + \mathbf{Ux} + \mathbf{b})\right\|_2 \leq \\ \frac{\left\|\overline{\mathbf{W}} - \mathbf{W}\right\|_2 \|\mathbf{Ux} + \mathbf{b}\|_2 + \left\|(\overline{\mathbf{U}} - \mathbf{U})\mathbf{x}\right\|_2 + \|\overline{\mathbf{b}} - \mathbf{b}\|_2}{(1-m)^2}\end{aligned} \tag{26}$$

Like former works [35], we we also introduce another lemma [33] here:

Lemma 2. *Let f_w be any predictor with parameters w, and let P denote any distribution on the parameters that are independent of the training data. Then, for any $\delta, \gamma > 0$, with probability $\geq 1 - \delta$ over the training data of size M, for any w, and any random perturbation u such that $\mathbb{P}\left[\max_x \|f_{w+u}(x) - f_w(x)\|_\infty < \frac{\eta}{4}\right] \geq \frac{1}{2}$, we have*

$$\mathcal{L}_0(f_w) \leq \hat{\mathcal{L}}_\eta f_w + 4\sqrt{\frac{KL(w+u||P) + \ln\frac{6M}{\delta}}{M-1}} \tag{27}$$

Then we can derive the perturbation bound for the reformulated OptEqs and our GEQ following former settings [35]. First, we also incorporate a fully connected layer at the end as we mentioned in the paper.

$$f^c_{geq}(\mathbf{x}) = \mathbf{W}_c f_{geq}(\mathbf{x}) + \mathbf{b}_c, f^c_{opteq}(\mathbf{x}) = \mathbf{W}_c f_{opteq}(\mathbf{x}) + \mathbf{b}_c. \tag{28}$$

Since the entries in the perturbations obeying the distribution os $\mathcal{N}(0, \sigma^2)$, we have that all the perturbations of weights $\|\Delta.\|$ are bounded by $\sigma\sqrt{2h\mathrm{ln}(24h)} : \omega$ with probability larger than $1/2$.

Since the only difference between OptEqs and monDEQ [48] is the weight parameterization, our reformulate OptEqs is parameterized by $\mathbf{W}_s = \mathbf{W}^\top\mathbf{W}$ while the monDEQ's weight parameter is parameterized by a series of weights $\mathbf{W}_{mondeq} = \mathbf{I} + \mathbf{A} + \mathbf{A}^\top + \mathbf{B} + \mathbf{B}^\top$. Therefore, the pertubation in $\|\Delta_{\mathbf{W}_s}\|$ is different from former analysis [35]. We have that:

$$\|\Delta_{\mathbf{W}_s}\|_2 = \|\Delta^\top_{\mathbf{W}}\mathbf{W} + \mathbf{W}\Delta^\top_{\mathbf{W}}\|_2 \leq 2\omega\mu \tag{29}$$

Then using the above lemma, we have that for all $\mathbf{x}$ with probability at least $1/2$.

$$\begin{aligned}\|\overline{f}^c_{opteq}(\mathbf{x}) - f^c_{opteq}(\mathbf{x})\|_2 &\leq \|\overline{\mathbf{W}}_c\overline{f}_{opteq}(\mathbf{x}) + \overline{\mathbf{b}}_c - \mathbf{W}_c f^c_{opteq}(\mathbf{x}) - \mathbf{b}\|_2 \\ &\leq \frac{2\mu^2\omega(B+1)}{(1-m)^2} + \frac{2\mu\omega(B+1)}{1-m} + \omega\end{aligned} \tag{30}$$

Setting $\sigma = \frac{\eta(1-m^2)}{4\sqrt{2h\mathrm{ln}(24h)}(2\mu^3(B+1)+2(1-m)\mu+(1-m)^2)}$ will make the above perturbation less than $\frac{\eta}{4}$. Then we have,

$$KL(\mathbf{W}_{\cdot} + \Delta_{\mathbf{W}_{\cdot}}|P) \leq \frac{\mathbf{B}_W}{2\sigma^2} = \frac{16h\ln(24h)(2\mu^3(B+1) + 2(1-m)\mu(B+1) + (1-m)^2)^2}{\eta^2(1-m)^4}\mathcal{B}_W \tag{31}$$

With the same choice of β's bound like former work [35],

$$\frac{\eta(1-m)}{2(B+1)} \leq \beta \leq \frac{\eta(1-m)\sqrt{M}}{2(B+1)}, \tag{32}$$

we can finally get the upper bound as our OptEqs bound as our proposition shows.

The difference between GEQ and OptEqs is that GEQ's can be viewed as multiplying scaler $\beta = e^{-\gamma\|\mathbf{Ux}+\mathbf{b}-\mathbf{Wz}\|_2^2}$ with depend on $\mathbf{x}$ since $\mathbf{z}$ is also depended on $\mathbf{x}$. Setting $\beta_{\max} = \max_{\mathbf{x}\in\mathcal{D}} \beta(\mathbf{x}) < 1$ and $\beta_{\min} = \min_{\mathbf{x}\in\mathcal{D}} \beta(\mathbf{x}) > c$. Assuming β changes a little with respect to the small perturbations on weights, we have:

$$\begin{aligned}
&\mathbf{z}^*(\mathbf{W},\mathbf{U},\mathbf{b}) = \sigma(-\beta\mathbf{W}_s\mathbf{z}^{(i)} + \beta\mathbf{W}(\mathbf{Ux}+\mathbf{b}))\\
&\|\mathbf{z}^*(\overline{\mathbf{W}},\overline{\mathbf{U}},\overline{\mathbf{b}})\|_2 \leq \frac{\beta_{\max}\mu\|\overline{\mathbf{W}}_s - \mathbf{W}_s\|_2\|\mathbf{Ux}+\mathbf{b}\|_2}{(1-\beta_{\max}m)^2} + \frac{\beta_{\max}(\|(\overline{\mathbf{WU}} - \mathbf{WU})\mathbf{x}\|_2 + \|\overline{\mathbf{Wb}} - \mathbf{Wb}\|_2)}{1-\beta_{\max}m}
\end{aligned} \tag{33}$$

With the same setting as above OptEqs, we have:

$$\|(\overline{\mathbf{WU}}-\mathbf{WU})\mathbf{x}\|_2 = \|\Delta_{\mathbf{W}}\overline{\mathbf{U}}-\mathbf{W}\Delta_{\mathbf{U}}\mathbf{x}\|_2 \leq 2\omega\mu\|(\overline{\mathbf{Wb}}-\mathbf{Wb})\mathbf{x}\|_2 = \|\Delta_{\mathbf{W}}\overline{\mathbf{b}}-\mathbf{W}\Delta_{\mathbf{b}}\mathbf{x}\|_2 \leq 2\omega\mu \tag{34}$$

Then we can obtain that:

$$\begin{aligned}
\|\overline{f}_{geq}^{c}(\mathbf{x}) - f_{kereq}^{c}(\mathbf{x})\|_2 &\leq \|\overline{\mathbf{W}}_c\overline{f}_{opteq}(\mathbf{x}) + \overline{\mathbf{b}}_c - \mathbf{W}_c f_{opteq}^{c}(\mathbf{x}) - \mathbf{b}\|_2\\
&\leq \frac{2\beta_{\max}\mu^3\omega(B+1)}{(1-\beta_{\max}m)^2} + \frac{2\mu^2\omega\beta_{max}(B+1)}{1-\beta_{max}m} + \frac{\mu\omega(B+1)}{1-\beta_{max}m} + \omega\\
&= \frac{(\beta_{\max}\mu) * (2\mu^2\omega(B+1))}{(1-\beta_{\max}m)^2} + \frac{(2\mu\beta_{max}+1) * (\mu\omega(B+1))}{1-\beta_{max}m} + \omega
\end{aligned} \tag{35}$$

With the same setting as above, we have:

$$\begin{aligned}
KL(\mathbf{W}_{\cdot} + \Delta_{\mathbf{W}_{\cdot}}|P) &\leq \frac{\mathbf{B}_W}{2\sigma^2}\\
&= T\mathcal{B}_W,
\end{aligned} \tag{36}$$

where T is defined as follows:

$$T = \frac{16h\ln(24h)(2(\beta_{\max}\mu)\mu^3(B+1) + 2(2\mu\beta_{\max}+1)(1-\beta_{\max}m)\mu(B+1) + (1-\beta_{\max}m)^2)^2}{\eta^2(1-\beta_{\max}m)^4} \tag{37}$$

then we can finally we can get the upper bound as our OptEqs bound as our proposition shows.

A.5 Proofs for proposition 3

Lipschitz constant is the minimal constant for f and $\forall\mathbf{x},\mathbf{y}$ suits the following equation:

$$\|f(\mathbf{x}) - f(\mathbf{y})\|_2 \leq L\|\mathbf{x}-\mathbf{y}\|_2. \tag{38}$$

Thereby, our analysis in Proposition 3 can be viewed as propose an upper bound for different models. In this section, we are going to prove the Lipschitz upper bounds for our GEQ and OptEqs. First, we restate the proposition as follows:

Proposition 3. *If norms for the inputs and outputs are bounded by B, the spectral norm for the weight parameter $\mathbf{W},\mathbf{U}$ of equilibrium models with ReLU activation are bounded by $\mu < 1$ to ensure convergence, then we have the conclusions as below:*

$$\|f_{geq}(\mathbf{x}_1) - f_{geq}(\mathbf{x}_1)\|_2 \leq L_{geq}\|\mathbf{x}_1 - \mathbf{x}_2\|_2 = \frac{\beta_{\max}\mu^2 + \sqrt{\gamma}B\mu^3}{1-\beta_{\max}\mu^2 - \sqrt{\gamma}B\mu^3}\|\mathbf{x}_1 - \mathbf{x}_2\|_2, \tag{39}$$

$$\|f_{opteq}(\mathbf{x}_1) - f_{opteq}(\mathbf{x}_2)\|_2 \leq L_{opteq}\|\mathbf{x}_1 - \mathbf{x}_2\|_2 = \frac{\mu}{1-\mu^2}\|\mathbf{x}_1 - \mathbf{x}_2\|_2, \tag{40}$$

where $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are input samples,* $f_{geq}(\mathbf{x}.)$ *and* $f_{opteq}(\mathbf{x}.)$ *denotes the equilibrium states for GEQ and OptEqs given input* $\mathbf{x}.$*, and* $\beta_{max} := \max_{\mathbf{x}\in\mathcal{D}} e^{-\gamma\|\mathbf{U}\mathbf{x}-\mathbf{W}\mathbf{z}\|_2^2} < 1$.

Proof. We first proof the inequality for OptEqs:

$$\begin{aligned}\|f_{opteq}(\mathbf{x}_1) - f_{opteq}(\mathbf{x}_2)\|_2 = \|\mathbf{z}_{\mathbf{x}_1} - \mathbf{z}_{\mathbf{x}_2}\|_2 &\leq \left\|\sigma\left(\mathbf{W}^\top\mathbf{W}\mathbf{z}_{\mathbf{x}_1} + \mathbf{U}\mathbf{x}_1\right) - \sigma\left(\mathbf{W}^\top\mathbf{W}\mathbf{z}_{\mathbf{x}_2} + \mathbf{U}\mathbf{x}_2\right)\right\|_2 \\ &\leq \left\|\mathbf{W}^\top\mathbf{W}(\mathbf{z}_{\mathbf{x}_1} - \mathbf{z}_{\mathbf{x}_2})\right\|_2 + \|\mathbf{U}(\mathbf{x}_1 - \mathbf{x}_2)\|_2 \\ &\leq \mu^2\|\mathbf{z}_{\mathbf{x}_1} - \mathbf{z}_{\mathbf{x}_2}\|_2 + \mu\|\mathbf{x}_1 - \mathbf{x}_2\|_2,\end{aligned} \tag{41}$$

where $\mathbf{z}_{\mathbf{x}_1}$ denotes the equilibrium states for OptEq with input $\mathbf{x}_1$, which means the following equation is satisfied:

$$\mathbf{z}_{\mathbf{x}_1} = \sigma\left(\mathbf{W}^\top\mathbf{W}\mathbf{z}_{\mathbf{x}_1} + \mathbf{U}\mathbf{x}_1\right). \tag{42}$$

Reformulating the equations, we can get:

$$\|f_{opteq}(\mathbf{x}_1) - f_{opteq}(\mathbf{x}_2)\|_2 \leq \frac{\mu}{1-\mu^2}\|\mathbf{x}_1 - \mathbf{x}_2\|_2. \tag{43}$$

Then for GEQ, we have:

$$\begin{aligned}\|\mathbf{z}_{\mathbf{x}_1} - \mathbf{z}_{\mathbf{x}_2}\|_2 \leq &\Big\|\sigma\left(e^{-\gamma\left\|\mathbf{W}\mathbf{z}_{\mathbf{x}_1} - \mathbf{U}\mathbf{x}_1\right\|_2^2}\left(-\mathbf{W}^\top\mathbf{W}\mathbf{z}_{\mathbf{x}_1} + \mathbf{W}^\top\mathbf{U}\mathbf{x}_1\right)\right) \\ &\quad -\sigma\left(e^{-\gamma\left\|\mathbf{W}\mathbf{z}_{\mathbf{x}_2} - \mathbf{U}\mathbf{x}_2\right\|_2^2}\left(-\mathbf{W}^\top\mathbf{W}\mathbf{z}_{\mathbf{x}_2} + \mathbf{W}^\top\mathbf{U}\mathbf{x}_2\right)\right)\Big\|_2 \\ \leq &\beta_{\max}\left\|\mathbf{W}^\top\mathbf{W}(\mathbf{z}_{\mathbf{x}_1} - \mathbf{z}_{\mathbf{x}_2})\right\|_2 + \left\|\mathbf{W}^\top\mathbf{U}(\mathbf{x}_1 - \mathbf{x}_2)\right\|_2 + B\mu^2\left|\beta_{\mathbf{x}_1} - \beta_{\mathbf{x}_2}\right|, \\ \leq &\beta_{\max}\mu^2\left\|\mathbf{z}_{\mathbf{x}_1} - \mathbf{z}_{\mathbf{x}_2}\right\|_2 + \beta_{\max}\mu^2\left\|\mathbf{x}_1 - \mathbf{x}_2\right\|_2 + B\mu^2\left|\beta_{\mathbf{x}_1} - \beta_{\mathbf{x}_2}\right|\end{aligned} \tag{44}$$

where $\mathbf{z}_{\mathbf{x}_1}$ denotes the equilibrium states for GEQ with input $\mathbf{x}_1$, and $\beta_{\mathbf{x}_1}$ is defined as follows:

$$\beta_{\mathbf{x}_1} = e^{-\gamma\left\|\mathbf{W}\mathbf{z}_{\mathbf{x}_1} - \mathbf{U}\mathbf{x}\right\|_2^2}. \tag{45}$$

Then with the mean value theorem, we have the following equations:

$$\begin{aligned}\left|\beta_{\mathbf{x}_1}\right. &\left. -\beta_{\mathbf{x}_2}\right| = \left|e^{-\gamma\left\|\mathbf{W}\mathbf{z}_{\mathbf{x}_1} - \mathbf{U}\mathbf{x}_2\right\|_2^2} - e^{-\gamma\left\|\mathbf{W}\mathbf{z}_{\mathbf{x}_1} - \mathbf{U}\mathbf{x}_2\right\|_2^2}\right| \\ &\leq \sqrt{\gamma}\left|\,\|\mathbf{W}\mathbf{z}_{\mathbf{x}_1} - \mathbf{U}\mathbf{x}_2\|_2 - \|\mathbf{W}\mathbf{z}_{\mathbf{x}_x} - \mathbf{U}\mathbf{x}_2\|_2\,\right| \cdot \max_{c\in\sqrt{\gamma}\left[\left\|\mathbf{W}\mathbf{z}_{\mathbf{x}_1} - \mathbf{U}\mathbf{x}_1\right\|_2, \left\|\mathbf{W}\mathbf{z}_{\mathbf{x}_2} - \mathbf{U}\mathbf{x}_2\right\|_2\right]} ce^{-c^2} \\ &\leq \sqrt{\gamma}\left|\,\|\mathbf{W}\mathbf{z}_{\mathbf{x}_2} - \mathbf{U}\mathbf{x}_2\|_2 - \|\mathbf{W}\mathbf{z}_{\mathbf{x}_2} - \mathbf{U}\mathbf{x}_2\|_2\,\right| \\ &\leq \sqrt{\gamma}\,\|\mathbf{W}(\mathbf{z}_{\mathbf{x}_1} - \mathbf{z}_{\mathbf{x}_2})\|_2 + \|\mathbf{U}(\mathbf{x}_1 - \mathbf{x}_2)\|_2 \\ &\leq \sqrt{\gamma}\mu\left(\|\mathbf{z}_{\mathbf{x}_1} - \mathbf{z}_{\mathbf{x}_2}\|_2 + \|\mathbf{x}_1 - \mathbf{x}_2\|_2\right)\end{aligned} \tag{46}$$

Then reformulating Eqn (44), we can finally obtain our proposition as below:

$$\|\mathbf{z}_{\mathbf{x}_1} - \mathbf{z}_{\mathbf{x}_2}\|_2 = \|f_{geq}(\mathbf{x}_1) - f_{geq}(\mathbf{x}_2)\|_2 \leq \frac{\beta_{\max}\mu^2 + \sqrt{\gamma}B\mu^3}{1 - \beta_{\max}\mu^2 - \sqrt{\gamma}B\mu^3}\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \tag{47}$$

□

A.6 Proofs for proposition 4

In this section, we are going to prove the output similarity bounds for our GEQ and OptEqs. First, we restate the proposition as follows:

Proposition 4. *If norms for the inputs and outputs are bounded by* B*, the spectral norm for the weight parameter* $\mathbf{W}$ *of equilibrium models with ReLU activation are bounded by* $\mu < 1$ *to ensure the convergence, and each row in* $\mathbf{U}$ *obeys the spherical Gaussian distributions* $\mathcal{N}(0, \mathbb{E}[U_i^2]\mathbf{I})$*. Then we have the following conclusions for the expectation of the output similarity for GEQ and OptEqs with respect to input* $\mathbf{x}_1, \mathbf{x}_2$ *as follows,*

$$\kappa_{geq}(\mathbf{x}_1, \mathbf{x}_2) \leq \overline{\kappa}_{geq} = \frac{\mu^2 D e^{-\gamma(\sigma_{\min}(\mathbf{U})^2\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2)}\mathbb{E}[U_i^2]\|\mathbf{x}_1\|\|\mathbf{x}_2\|\left(\sin\theta_0 + (\pi - \theta_0)\cos\theta_0\right)}{2\pi(1 - \beta_{\max}\mu^2)^2}, \tag{48}$$

$$\kappa_{opteq}(\mathbf{x}_1, \mathbf{x}_2) \leq \overline{\kappa}_{opteq} = \frac{\mathbb{E}[U_i^2]\|\mathbf{x}_1\|\|\mathbf{x}_2\|\left(\sin\theta_0 + (\pi - \theta_0)\cos\theta_0\right)}{2\pi(1 - \mu^2)^2}, \tag{49}$$

where $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are input samples,* $D = e^{\gamma B\|\mathbf{W}\|_2^2}$ *and* $\beta_{max} := \max_{\mathbf{x}\in\mathcal{D}} e^{-\gamma\|\mathbf{U}\mathbf{x}-\mathbf{W}\mathbf{z}\|_2^2} < 1$. $\theta_0 = \cos^{-1}(\frac{\langle \mathbf{x}_1,\mathbf{x}_2\rangle}{\|\mathbf{x}_1\|\|\mathbf{x}_2\|})$ *is defined as the angle between the samples.*

Proof. Before the proof, we first introduce the following lemma:

Lemma 3. *[45] If* $\mathbf{U}$ *obeys the spherical Gaussian distributions of variance* $\mathbb{E}[U_i^2]$ *and mean* 0*, then the expectation of the Similarity for the one-layer Neural Network* $\sigma(\mathbf{U}\mathbf{x})$ *is:*

$$\kappa_{NN}(\mathbf{x}_1,\mathbf{x}_2) = \frac{\mathbb{E}[U_i^2]\|\mathbf{x}_1\|\|\mathbf{x}_2\|}{2\pi}\left(\sin\theta_0 + (\pi-\theta_0)\cos\theta_0\right) \tag{50}$$

where $\theta_0 = \cos^{-1}(\frac{\langle \mathbf{x},\mathbf{y}\rangle}{\|\mathbf{x}\|\|\mathbf{y}\|})$.

Letting $m := \|\mathbf{W}_s\|_2 = \|\mathbf{W}^\top\mathbf{W}\|_2 < 1$ and $\mu := \|\mathbf{W}\|_2 < 1$ as our assumptions demonstrate and neglecting the bias $\mathbf{b}$ for convenience. Then for our reformulated OptEqs, we have:

$$\begin{aligned} \mathbf{z}_{\mathbf{x}_1}^{*\top}\mathbf{z}_{\mathbf{x}_2} &\le \sigma(\mathbf{W}_s\mathbf{z}_{\mathbf{x}_1})^\top\mathbf{z}_{\mathbf{x}_2} + \sigma(\mathbf{U}\mathbf{x}_1)^\top\mathbf{z}_{\mathbf{x}_2} \\ &\le \mu^2\sigma(\mathbf{z}_{\mathbf{x}_1})^\top\mathbf{z}_{\mathbf{x}_2} + \sigma(\mathbf{U}\mathbf{x}_1)^\top\mathbf{z}_{\mathbf{x}_2} \end{aligned} \tag{51}$$

And we have

$$\begin{aligned} \sigma(\mathbf{U}\mathbf{x}_1)^\top\mathbf{z}_{\mathbf{x}_2} &\le \sigma(\mathbf{U}\mathbf{x}_1)^\top\sigma(\mathbf{W}\mathbf{z}_{\mathbf{x}_2}) + \sigma(\mathbf{U}\mathbf{x}_1)^\top\sigma(\mathbf{U}\mathbf{x}_2) \\ &\le \mu^2\sigma(\mathbf{U}\mathbf{x}_1)^\top\mathbf{z}_{\mathbf{x}_2} + \sigma(\mathbf{U}\mathbf{x}_1)^\top\sigma(\mathbf{U}\mathbf{x}_2) \end{aligned} \tag{52}$$

Then

$$\begin{aligned} \sigma(\mathbf{U}\mathbf{x}_1)^\top\mathbf{z}_{\mathbf{x}_2} &\le \frac{\sigma(\mathbf{U}\mathbf{x}_1)^\top\sigma(\mathbf{U}\mathbf{x}_2)}{1-\mu^2} \\ \mathbf{z}_{\mathbf{x}_1}^{*}{}^\top\mathbf{z}_{\mathbf{x}_2} &\le \frac{\sigma(\mathbf{U}\mathbf{x}_1)^\top\sigma(\mathbf{U}\mathbf{x}_2)}{(1-\mu^2)^2} \end{aligned} \tag{53}$$

Therefore, we can conclude

$$\kappa_{opteq}(\mathbf{x}_1,\mathbf{x}_2) \le \frac{\mathbb{E}[U_i^2]\|\mathbf{x}_1\|\|\mathbf{x}_2\|\left(\sin\theta_0 + (\pi-\theta_0)\cos\theta_0\right)}{2\pi(1-\mu^2)^2} \tag{54}$$

For GEQ, we set $\beta_{\mathbf{x}} = e^{-\gamma\|\mathbf{U}\mathbf{x}-\mathbf{W}\mathbf{z}\|_2^2}$ and $\beta_{max} := \max\limits_{\mathbf{x}\in\mathcal{D}} e^{-\gamma\|\mathbf{U}\mathbf{x}-\mathbf{W}\mathbf{z}\|_2^2} < 1$

$$\begin{aligned} \mathbf{z}_{\mathbf{x}_1}^{*\top}\mathbf{z}_{\mathbf{x}_2} &\le \beta_{\max}\sigma(\mathbf{W}_s\mathbf{z}_{\mathbf{x}_1})^\top\mathbf{z}_{\mathbf{x}_2} + \beta_{\mathbf{x}}\|\mathbf{W}\|_2\sigma(\mathbf{U}\mathbf{x}_1)^\top\mathbf{z}_{\mathbf{x}_2} \\ &\le \beta_{\max}\mu^2\sigma(\mathbf{z}_{\mathbf{x}_1})^\top\mathbf{z}_{\mathbf{x}_2} + \beta_{\mathbf{x}}\mu\sigma(\mathbf{U}\mathbf{x}_1)^\top\mathbf{z}_{\mathbf{x}_2} \end{aligned} \tag{55}$$

Like OptEqs, we obtain the following equations for GEQ:

$$\mathbf{z}_{\mathbf{x}_1}^{*\top}\mathbf{z}_{\mathbf{x}_2} \le \frac{\mu^2\beta_{\mathbf{x}_1}\beta_{\mathbf{x}_2}\sigma(\mathbf{U}\mathbf{x}_1)^\top\sigma(\mathbf{U}\mathbf{x}_2)}{(1-\beta_{\max}\mu^2)^2} \tag{56}$$

Therefore,

$$\kappa_{geq}(\mathbf{x}_1,\mathbf{y}_2) \le \frac{\mu^2\beta_{\mathbf{x}_1}\beta_{\mathbf{x}_2}\mathbb{E}[U_i^2]\|\mathbf{x}_1\|_2\|\mathbf{x}_2\|_2\left(\sin\theta_0 + (\pi-\theta_0)\cos\theta_0\right)}{2\pi(1-\beta_{\max}\mu^2)^2} \tag{57}$$

And we have,

$$\begin{aligned} \beta_{\mathbf{x}_1}\beta_{\mathbf{x}_2} &= e^{-\gamma(\|\mathbf{W}\mathbf{z}_{\mathbf{x}_1}-\mathbf{U}\mathbf{x}_1\|_2^2+\|\mathbf{W}\mathbf{z}_{\mathbf{x}_2}-\mathbf{U}\mathbf{x}_2\|_2^2)} \\ &\le e^{-\gamma/2(\|\mathbf{W}(\mathbf{z}_{\mathbf{x}_1}-\mathbf{z}_{\mathbf{x}_2})-\mathbf{U}(\mathbf{x}_1-\mathbf{x}_2)\|_2^2)} \\ &\le De^{-\frac{\gamma}{4}(\sigma_{\min}(\mathbf{U})^2\|\mathbf{x}_1-\mathbf{x}_2\|_2^2)}, \end{aligned} \tag{58}$$

where $D = e^{\gamma\|\mathbf{W}\|_2^2 B}$, the latter two inequality is acquired by Jensen's inequality. Then we have,

$$\kappa_{geq}(\mathbf{x}_1,\mathbf{x}_2) \le \frac{\mu^2 De^{-\frac{\gamma}{4}(\sigma_{\min}(\mathbf{U})^2\|\mathbf{x}_1-\mathbf{x}_2\|_2^2)}\mathbb{E}[U_i^2]\|\mathbf{x}_1\|_2\|\mathbf{x}_2\|_2\left(\sin\theta_0 + (\pi-\theta_0)\cos\theta_0\right)}{2\pi(1-\beta_{\max}\mu^2)^2} \tag{59}$$

□

A.7 Experiment Settings

A.7.1 Experiments on CIFAR

For our GEQ, we parallel 6 branches with each branch taking the scale of $32, 16, 8, 8, 4, 4$ and using the average fusion method for branches' fusion. The output channels for 6 branches are all 256 or 320 but the mid-channel number(output channel for weight $\mathbf{U}$ and $\mathbf{W}$) for the six branches are $64, 128, 128, 128, 256, 256$ or $80, 160, 160, 160, 320, 320$ with patch size 2 and c splitting is 8. And the inner MLP inner $\mathbf{W}_h$ output 64 hidden dimension for each patch. We use the SGD [23] optimizer with momentum and step learning rate schedule for all the models. We also use RandomAug for all the models for comparison.

A.7.2 Experiments on ImageNette and ImageNet-100

We take the input scale as 256 for all models. For our GEQ, we parallel 6 branches with each branch taking the scale of $64, 32, 16, 16, 8, 8$ after two downsampling convolutions and using the average fusion method for branches' fusion. The output channels for 6 branches are all 256 or 384 but the mid-channel number(output channel for weight $\mathbf{U}$ and $\mathbf{W}$) for the six branches are $32, 64, 128, 128, 256, 256$ or $48, 96, 192, 192, 384, 384$ with patch size 2 and c splitting is 8. And the inner MLP inner $\mathbf{W}_h$ output 128 hidden dimension for each patch. We use the SGD optimizer with momentum and step learning rate schedule for all the models. We also use RandomAug for all the models for comparison.